

## Re: RFC: libkse\*.a in 7.0

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2007-12/msg00060.html>

---

- *From:* Alexander Leidinger <[Alexander@xxxxxxxxxxxxx](mailto:Alexander@xxxxxxxxxxxxx)>
  - *Date:* Tue, 11 Dec 2007 08:02:16 +0100
- 

Quoting David Schultz <[das@xxxxxxxxxxxxx](mailto:das@xxxxxxxxxxxxx)> (from Mon, 10 Dec 2007 17:38:38 -0500):

On Mon, Dec 10, 2007, Alexander Leidinger wrote:

Running Solaris 8/9 programs is not supported by SUN on Solaris 10. It works in some cases, but it doesn't work in some other cases.

That's not true. It is supported. See:

<http://www.sun.com/software/solaris/programs/abi/>

<http://www.sun.com/software/solaris/programs/abi/sag.xml>

In theory, a SunOS 5.0 app will still work in SunOS 5.10

The important part is the "theory" word...

I work in the office of SUN in Luxembourg, and one of our ideas for a client was to run a Solaris 8/9 in a zone of a Solaris 10 as a replacement for machines with Solaris 8/9. As we have a service contract with our client, we have to take some business constraints into account. And one of those business constraints is that Solaris 8/9 in a zone of Solaris 10 is not supported, as the kernel interface (syscalls) changed in an incompatible way.

Of course, in practice, perfect binary compatibility is too much to ask for. It's possible to write programs that notice that

So far we handled this good in FreeBSD.

different releases aren't bug-for-bug compatible, and if you statically link your binary or use unsupported ABIs, you break their guarantee. But that's orthogonal to my original point.

And now

some people work on using BrandZ (if you know nothing about it, it's sort of like our technology used to do our linuxulator or freebsd32 on

Re: RFC: libkse\*.a in 7.0

amd64; that's not accurate, but is good enough for the point I want to make) to provide a Solaris 10 container (think about it as a jail on steroids) with an Solaris X (X < 10) image, so that people can install a Solaris 10 host and run Solaris X in it (like our linuxulator in a jail, but not as flexible as our linuxulator, theirs can not run on the main system like ours can).

Right, having the linuxulator in the kernel is all but unavoidable. But for old FreeBSD apps running on newer versions of FreeBSD, we can do better, and a library-based approach is easier to maintain and less prone to security problems.

It's not running only old apps on a new system, it's running the userland of an old system in a jail of a new system. That's what I'm concerned about (and works currently as we took care about maintaining compatibility in the kernel) and that's what you can not handle with a library-based approach.

Bye,  
Alexander.

---

You get what you pay for.  
-- Gabriel Biel

<http://www.Leidinger.net> Alexander @ Leidinger.net: PGP ID = B0063FE7  
<http://www.FreeBSD.org> netchild @ FreeBSD.org : PGP ID = 72077137

---

freebsd-arch@xxxxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@xxxxxxxxxxxxx"